METHODS, APPARATUS AND ARTICLES-OF-MANUFACTURE FOR PROVIDING ALWAYS-LIVE DISTRIBUTED COMPUTING

By: James Bernardin and Peter Lee

CROSS-REFERENCE TO RELATED APPLICATION

This application is a continuation-in-part of U.S. Patent Application S/N 09/583,244, filed 5/31/00, by the inventors herein ("the '244 application"), which prior application is incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates generally to the fields of distributed computing methods, computer-assisted business methods, and systems and articles-of-manufacture for implementing such methods. More particularly, the invention relates to computer-based methods, apparatus and articles-of-manufacture for providing "always-live" (i.e., substantially continuously active and uninterrupted) distributed computing services in a network-based computing environment.

BACKGROUND OF THE INVENTION

Methods for providing distributed computing in network-based computing environments (such as the Internet) are known. One widely-publicized effort was the so-called SETI@home (Search for Extra-Terrestrial Intelligence) project, in which large numbers of Internet-connected computers were used to

process radio-telescope data, in an effort to identify patterns indicative of intelligent life. Other examples are described in U.S. Patent Nos. 5,964,832 ("USING NETWORKED REMOTE COMPUTERS TO EXECUTE COMPUTER PROCESSING TASKS AT A PREDETERMINED TIME"), 6,098,091 ("METHOD AND SYSTEM INCLUDING A CENTRAL COMPUTER THAT ASSIGNS TASKS TO IDLE WORKSTATIONS USING AVAILABILITY SCHEDULES AND COMPUTATIONAL CAPABILITIES") and 6,112,243 ("METHOD AND APPARATUS FOR ALLOCATING TASKS TO REMOTE NETWORKED PROCESSORS"), all owned by Intel Corporation. Still another example is disclosed in the earlier-filed '244 application by the inventors herein. (Note, however, that the '244 application is <u>not</u> prior art to the present invention.)

Generally speaking, the primary object of Internet-based distributed computing systems is to exploit the vast computational resources that sit idle for much of the 24-hour day on computer networks around the world. Although some success has been achieved, prior-art systems still have problems that limit their usefulness in real-world applications.

One particularly-troublesome aspect of the prior-art systems is their inability guarantee timely results. While it may be no problem for the SETI@home researchers to wait days or weeks for results from a particular data set, commercial customers simply cannot afford to have overnight processing jobs run unexpectedly into the next business day. Therefore, in order to realize the full commercial

potential of network-based distributed computing, it is necessary to ensure that the clients' work gets processed in a substantially continuous and uninterrupted manner, so that a service provider can assure his/her client that assigned work will be completed in within a commercially-reasonable time period (e.g., an hour, four hours, eight hours, etc.).

OBJECTS AND DESCRIPTION OF THE INVENTION

In light of the above, a first general object of the invention relates to computer-based methods, apparatus and articles-of-manufacture that facilitate an always-live distributed computing system.

A second general object of the invention relates to computer-based methods, apparatus and articles-of-manufacture that provide substantially continuous monitoring of worker processor activity and/or task progress in a distributed computing environment.

A third general object of the invention relates to computer-based methods, apparatus and articles-of-manufacture that provide prompt alerts of worker processor status changes that can affect the always-live operation of a network-based distributed computing system.

A fourth general object of the invention relates to computer-based methods, apparatus and articles-of-manufacture for providing reliable and/or predictable resource deployment and processing activity in a wide-area network

based distributed computing system.

These, as well as other objects and advantages of the present invention, will become apparent in light of the following description, which details, by way of example, various aspects and features of the present invention.

Accordingly, generally speaking, and without intending to be limiting, one aspect of the invention relates to a method for operating a distributed computing system, the system including a multiplicity of network-connected worker processors and at least one supervisory processor, the supervisory processor configured to assign tasks to, and monitor the status of, the worker processors, the method comprising: assigning tasks to a plurality of the worker processors by sending task-assignment messages, via the network, from the at least one supervisory processor to the plurality of worker processors; and monitoring, on a substantially continuous basis, the status of at least each of the plurality of assigned worker processors until each processor completes its assigned task. Monitoring, on a substantially continuous basis, the status of at least each of the plurality of assigned worker processors may involve receiving status messages from at least each of the plurality of assigned worker processors until each processor completes its assigned task. Monitoring, on a substantially continuous basis, the status of at least each of the plurality of worker processors may also involve detecting abnormalities in the operation of the plurality of assigned worker processors, and/or their associated network connections, by detecting an absence of expected status message(s)

received by the at least one supervisory processor. Detection of an absence of expected status message(s) received by the at least one supervisory processor may be repeated at least once every ten minutes, once every five minutes, once every two minutes, once each minute, once every thirty seconds, once every ten seconds, once every second, once every tenth of a second, once every hundredth of a second, once each millisecond, or at whatever interval is needed to assure the continuity-of-service demanded by the client. Monitoring, on a substantially continuous basis, the status of at least each of the plurality of assigned worker processors may also involve detecting the presence of non-assigned-task-related activity on the worker processors. Detecting the presence of non-assigned-task-related activity on the worker processors may involve running an activity monitor program on each of the assigned worker processors. The activity monitor programs running on each of the assigned worker processors may behave substantially like screen saver programs. The activity monitory programs running on each of the assigned worker processors may send, in response to detection of keyboard activity (or mouse activity, pointer activity, touchscreen activity, voice activity, local execution of substantial non-assigned-task-related processes, or any combination thereof), a message to at least one of the at least one supervisory processor(s). Detecting the presence of non-assigned-task-related activity on the worker processors may also involve determining, in response to an activity monitor message received by at least one of the at least one supervisory of the processor(s),

that at least one of the assigned worker processors is undertaking non-assigned-task-related activity. The activity monitor message may be generated by an activity monitor program running on one of the assigned worker processors.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to a method for operating an always-live distributed computing system, comprising: providing a pool of worker processors, each having installed worker processor software, and each connected to an always-on, peer-to-peer computer network; providing at least one supervisory processor, also connected to the always-on, peer-to-peer computer network; using the at least one supervisory processor to monitor, on a substantially continuous basis, the status of worker processors expected to be engaged in the processing of assigned tasks; and using the at least one supervisory processor to reassign tasks, as needed, to achieve substantially uninterrupted processing of assigned tasks. Providing a pool of worker processors may further involve ensuring that each of the worker processors is linked to the always-on, peer-to-peer computer network through a high-bandwidth connection having, for example, a data rate of least 100 kilobits/sec, 250 kilobits/sec, 1 megabit/sec, 10 megabits/sec, 100 megabits/sec, 1 gigabit/sec, or whatever particular bandwidth may be demanded by the client's needs (e.g., required throughput and data intensiveness of the application). Using the at least one supervisory processor to monitor the status of worker processors expected to be engaged in the processing of assigned tasks may involve sending a

status-request message to, and receiving a return acknowledgement from, each worker processor that is expected to be engaged in the processing of assigned tasks. The process of sending a status-request message to, and receiving a return acknowledgement from, each worker processor that is expected to be engaged in the processing of assigned tasks is preferably repeated at least once every minute, second, tenth of a second, hundredth of a second, millisecond or other interval, as needed to meet client requirements. Using the at least one supervisory processor to monitor the status of worker processors expected to be engaged in the processing of assigned tasks may also involve periodically checking to ensure that a heartbeat message has been received, within a preselected frequency interval, from each worker processor that is expected to be engaged in the processing of assigned tasks. The preselected frequency interval may be set at or less than one minute, ten seconds, one second, one tenth of a second, one hundredth of a second, one millisecond, or other appropriate value, as needed. Using the at least one supervisory processor to reassign tasks, as needed, to achieve substantially uninterrupted processing of assigned tasks may also involve: detecting aberrant behavior among the worker processors expected to be engaged in the processing of assigned tasks; and reassigning tasks expected to be completed by the aberrant-behaving worker processor(s) to other available processor(s) in the worker processor pool.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to a method for operating a network-connected processor as a processing element in a distributed processing system, the method comprising: installing software that enables the network-connected processor to receive tasks from, and provide results to, one or more independent, network-connected resource(s); and using the software installed on the network-connected processor to provide substantially continuous status information to an independent, network-connected resource. Using the software installed on the network-connected processor to provide substantially continuous status information to an independent, network-connected resource may involve sending a heartbeat message to the independent, network-connected resource at least once every second, tenth of a second, hundredth of a second, millisecond, etc. Using the software installed on the network-connected processor to provide substantially continuous status information to an independent, network-connected resource may also involve responding to status-request messages, received from the independent, network-connected resource, within a predetermined response time, such as one second, one tenth of a second, one hundredth of a second, one millisecond, etc. Using the software installed on the network-connected processor to provide substantially continuous status information to an independent, network-connected resource may also involve sending, in response to a change in status of the network-connected processor, a status-update message to the independent,

network-connected resource within a preselected update interval, such as one second, one tenth of a second, one hundredth of a second, one millisecond, etc. The change in status that initiates the sending of a status-update message may include any local activity indicator (such as keyboard activity, other processes in the process queue, etc.) that indicates additional demand for the processing resources of the network-connected processor.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to a distributed computing system comprising: a multiplicity of worker processors; at least one supervisory processor, configured to assign tasks to, and monitor the status of, the worker processors; an always-on, peer-to-peer computer network linking the worker processors and the supervisory processor(s); and at least one of the at least one supervisory processor(s) including a monitoring module, which monitors the status of worker processors expected to be executing assigned tasks, so as to ensure that the distributed computing system maintains always-live operation. The monitoring module may receive status messages from at least each of the worker processors expected to be executing assigned tasks. The monitoring module may be used to detect abnormalities in the operation of the worker processors expected to be executing assigned tasks, and/or their associated network connections, by, for example, detecting an absence of expected status messages received from the worker processors. The monitoring module may repeatedly check for an absence of expected status messages at a

frequency of at least once each minute, at least once every ten seconds, at least once each second, at least once every tenth of a second, etc. The monitoring module may also be used to detect the presence of non-assigned-task-related activity on the worker processors expected to be executing assigned tasks. Activity monitor programs may be run on each of the worker processors expected to be executing assigned tasks. The activity monitor programs comprise screensaver programs. The activity monitor programs may be configured to detect one or more of the following types of non-assigned-task-related activity: keyboard activity; mouse activity; pointer activity; touchscreen activity; voice activity; and execution of substantial non-assigned-task-related processes.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to an always-live distributed computing system, comprising: a pool of worker processors, each having installed worker processor software, and each connected to an always-on, peer-to-peer computer network; and at least one supervisory processor, also connected to the always-on, peer-to-peer computer network, and configured to assign tasks to the worker processors, monitor, on a substantially continuous basis, the status of worker processors expected to be engaged in the processing of assigned tasks and reassign tasks, as needed, to achieve substantially uninterrupted processing of assigned tasks. The computer network may have a bandwidth of at least 250 kilobits/second, at least 1 megabit/second, etc. The at least one supervisory processor may monitor the status

of worker processors expected to be engaged in the processing of assigned tasks by sending a status-request message to, and receiving a return acknowledgement from, each worker processor that is expected to be engaged in the processing of assigned tasks. Such status-request message(s) may be sent at a frequency of at least once every 10 seconds, at least once each second, at least twenty times each second, etc. The at least one supervisory processor may monitor the status of worker processors expected to be engaged in the processing of assigned tasks by periodically checking to ensure that a heartbeat message has been received, within a preselected frequency interval, from each worker processor that is expected to be engaged in the processing of assigned tasks. The preselected frequency interval may be, for example, one second, one tenth of a second, one hundredth of a second, one millisecond, etc.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to a processing element for use in a distributed processing system, the processing element comprising: at least one processor; memory; at least one high-bandwidth interface to a computer network; and worker processor software, configured to receive tasks via the high-bandwidth interface and to provide substantially continuous status information via the high-bandwidth interface. The substantially continuous status information may be provided by sending periodic heartbeat messages. The substantially continuous status information may also be provided by sending prompt responses to received

status-request messages. The substantially continuous status information may also be provided by promptly sending a status-update message in response to changes in status.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to article(s)-of-manufacture for use in connection with a network-based distributed computing system, the article(s)-of-manufacture comprising at least one computer-readable medium containing instructions which, when executed, cause: assignment of tasks to a plurality of worker processors via the network; and monitoring, on a substantially continuous basis, of the status of at least each of the plurality of assigned worker processors until each such processor completes its assigned task.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to article(s)-of-manufacture for use in connection with an always-live distributed computing system, the article(s)-of-manufacture comprising at least one computer-readable medium containing instructions which, when executed, cause: a pool of worker processors to install worker processor software provided via an always-on, peer-to-peer computer network; provide communication paths between the worker processors and at least one supervisory processor via the always-on, peer-to-peer computer network; cause the at least one supervisory processor to monitor, on a substantially continuous basis, the status of worker processors expected to be engaged in the processing of assigned tasks; and

cause the at least one supervisory processor to reassign tasks, as needed, to achieve substantially uninterrupted processing of assigned tasks.

Again, generally speaking, and without intending to be limiting, another aspect of the invention relates to article(s)-of-manufacture for use in connection with a processing element constituting a part of a distributed computing system, the article(s)-of-manufacture comprising at least one computer-readable medium containing instructions which, when executed, cause: worker processor software to be installed that permits the processing element to receive tasks from, and provide results to, one or more independent, network-connected resource(s); and the installed worker processor software to be executed and provide substantially continuous status information to one or more of the independent, network-connected resource(s).

Still further aspects of the invention relate to alternative combinations, sub-combinations, supplemental combinations and/or permutations of the various above-described elements and features, as well as those elements and features described in the incorporated '244 application, consistent with or in furtherance of the objects and spirit of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Various aspects, features and advantages of the instant invention are depicted in the accompanying set figures, which is intended to be illustrative, rather

than limiting, and in which:

FIG. 1 depicts an exemplary network-based distributed processing system in which the present invention may be employed; and,

FIG. 2 contains a flowchart illustrating the operation of an exemplary always-live distributed processing system in accordance with the invention.

DESCRIPTION OF AN EXEMPLARY EMBODIMENT

Referring initially to FIG. 1, which depicts an exemplary context in which the method(s), apparatus and/or article(s)-of-manufacture of the invention may be applied, a computer network **1** is shown connecting a plurality of processing resources. (Although, for clarity, only six processing resources are shown in FIG. 1, the invention is preferably deployed in networks connecting hundreds, thousands, tens of thousands or greater numbers of processing resources.) Computer network **1** may utilize any type of transmission medium (e.g., wire, coax, fiber optics, RF, satellite, etc.) and any network protocol. However, in order to realize the principal benefit(s) of the present invention, computer network **1** should provide a relatively high bandwidth (e.g., at least 100 kilobits/second) and preferably, though not necessarily, should provide an "always on" connection to the processing resources involved in distributed processing activities.

Still referring to FIG. 1, one or more supervisory processor(s) **13** may communicate with a plurality of worker processors **10** via computer network **1**. Supervisory processor(s) **13** perform such tasks as:

- accepting job(s) from clients;
- assigning/reassigning tasks to (or among) worker processors;
- managing pools of available worker processors;
- monitoring the status of worker processors;
- monitoring the status of network connections;
- monitoring the status of job and task completions; and/or,
- resource utilization tracking, timekeeping and billing.

Still referring to FIG. 1, the depicted plurality **13** of worker processors **11** and **12** may operate collaboratively as a group, independently (e.g., each handing different job(s), task(s) and/or worker processor pool(s)) and/or redundantly (thus providing enhanced reliability). However, to realize a complete distributed processing system in accordance with the invention, only a single supervisory processor (e.g., **11** or **12**) is needed.

Still referring to FIG. 1, plurality **10** of worker processors illustratively comprises worker processors **2**, **4**, **6** and **8**, each connected to computer network **1** through network connections **3**, **5**, **7** and **9**, respectively. These worker processors communicate with supervisory processor(s) **13** via network **1**, and preferably include worker processor software that enables substantially continuous monitoring of

worker processor status and/or task execution progress by supervisory processor(s) **13**.

Referring now to FIG. 2, which depicts an exemplary "always-live" task monitoring/management process, a received job request **20** is initially assigned **21** to a plurality of available worker processors. Then, until the client's job is completed, processor(s) working on assigned task(s) are continuously monitored to ensure that the job is completed in a substantially uninterrupted (or "always live") manner. In particular, a monitoring module repeatedly asks whether all assigned tasks have been completed **22**. If so, then the job is complete, and results can be reported **23**. If not, then the monitoring module inquires about the status **24** of processor(s) expected to be working on not-yet-completed tasks. If potential bottlenecks are discovered, affected task(s) are immediately reassigned **25** to ensure that the system remains "live" and the client's work gets completed in a timely manner. This process is repeated with a frequency sufficient to ensure that worker processor problems will not cause undue delay is completing the overall job.

While the foregoing has described the invention by recitation of its various aspects/features and an illustrative embodiment thereof, those skilled in the art will recognize that alternative elements and techniques, and/or combinations and sub-combinations of the described elements and techniques, can be substituted for, or added to, those described herein. The present invention, therefore, should not be limited to, or defined by, the specific apparatus, methods, and articles-of-

manufacture described herein, but rather by the appended claims, which are intended to be construed in accordance with well-settled principles of claim construction, including, but not limited to, the following:

Limitations should not be read from the specification or drawings into the claims (e.g., if the claim calls for a "chair," and the specification and drawings show a rocking chair, the claim term "chair" should not be limited to a rocking chair, but rather should be construed to cover any type of "chair").

The words "comprising," "including," and "having" are always open-ended, irrespective of whether they appear as the primary transitional phrase of a claim, or as a transitional phrase within an element or sub-element of the claim (e.g., the claim "a widget comprising: A; B; and C" would be infringed by a device containing 2A's, B, and 3C's; also, the claim "a gizmo comprising: A; B, including X, Y, and Z; and C, having P and Q" would be infringed by a device containing 3A's, 2X's, 3Y's, Z, 6P's, and Q).

The indefinite articles "a" or "an" mean "one or more"; where, instead, a purely singular meaning is intended, a phrase such as "one," "only one," or "a single," will appear.

Where the phrase "means for" precedes a data processing or

manipulation "function," it is intended that the resulting means-plus-function element be construed to cover any, and all, computer implementation(s) of the recited "function" using any standard programming techniques known by, or available to, persons skilled in the computer programming arts.

A claim that contains more than one computer-implemented means-plus-function element should not be construed to require that each means-plus-function element must be a structurally distinct entity (such as a particular piece of hardware or block of code); rather, such claim should be construed merely to require that the overall combination of hardware/software which implements the invention must, as a whole, implement at least the function(s) called for by the claims.